TreeKs : un foncteur pour faire passer les domaines numériques à l'échelle

Stage de master 2 réalisé sous la direction d'Antoine Miné
École normale supérieure, équipe ABSTRACTION

Mehdi Bouaziz

Mardi 7 septembre 2010

Motivation

Analyse statique numérique :

- découverte automatique et statique de propriétés numériques sur les variables d'un programme

Applications :

- vérification statique de programmes (exemple : Astrée)
- découverte et preuve d'invariants
- optimisation de programmes

Cadre : domaines numériques abstraits

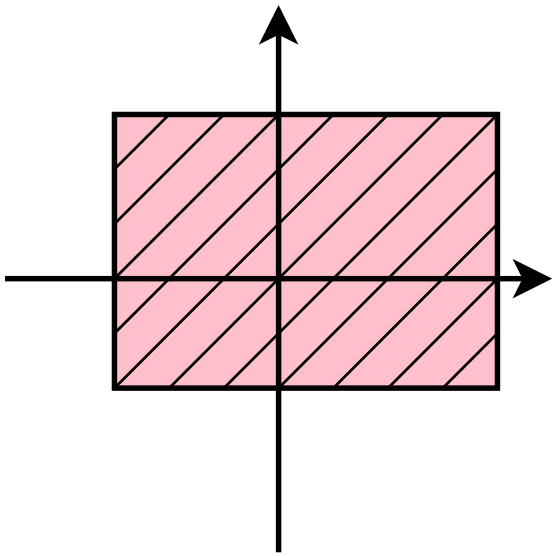
L'interprétation abstraite [Cousot Cousot 77] définit un cadre formel de l'approximation sûre de sémantiques.

Un domaine numérique abstrait est la donnée de :

- un ensemble $\mathcal{D}_{\mathcal{V}}$ de valeurs abstraites représentables sur un ordinateur,
- une concrétisation $[\![.]\!] : \mathcal{D}_{\mathcal{V}} \longrightarrow \mathcal{P}(\mathcal{V} \mapsto \mathbb{Q})$,
- un algorithme de comparaison $\sqsubseteq^{\mathcal{D}_{\mathcal{V}}}$ des éléments abstraits,
- des algorithmes efficaces et sûrs pour les opérateurs abstraits : intersection $\sqcap^{\mathcal{D}_{\mathcal{V}}}$, union $\sqcup^{\mathcal{D}_{\mathcal{V}}}$, projection $\exists^{\mathcal{D}_{\mathcal{V}}}$, ...

Domaines numériques abstraits : exemples

Intervalles [Cousot Cousot 76]

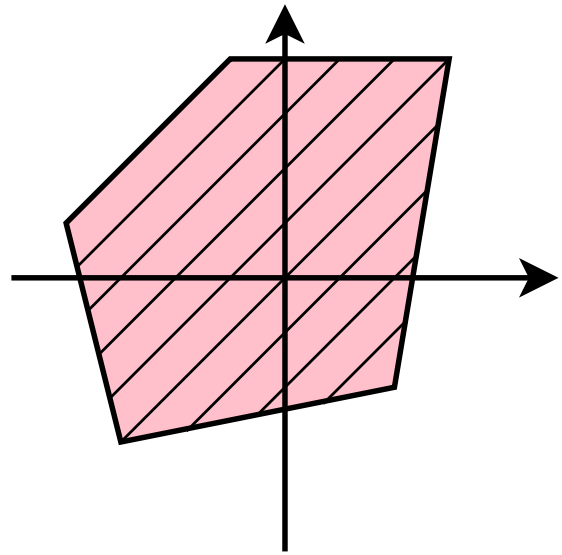


$\bigwedge_i a_i \leq X_i \leq b_i$

Non relationnel

Coût linéaire

Polyèdres [Cousot Halbwachs 78]



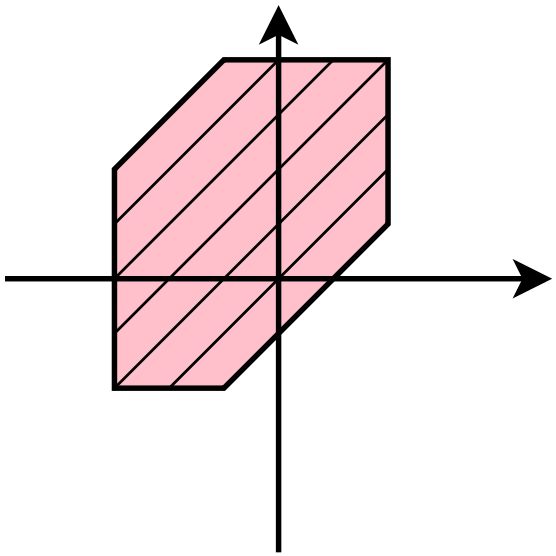
$\bigwedge_j \sum_i a_{ij} X_i \leq b_j$

Relationnel et très précis

Coût exponentiel en pire cas

Domaines numériques abstraits faiblement relationnels

Zones [Miné 01]



$\bigwedge_{ij} X_i - X_j \leq c_{ij}$

Faiblement relationnel

Coût cubique

Octogones [Miné 01]

$\bigwedge_{ij} \pm X_i \pm X_j \leq c_{ij}$

Coût cubique

Logaèdres [Howe King 09]

$\bigwedge_{ij} \pm 2^{a_i} X_i \pm 2^{b_j} X_j \leq c_{ij}$

Coût cubique

TVPI [Simon King Howe 02]

$\bigwedge_{ij} a_i X_i + b_j X_j \leq c_{ij}$

Coût quasi-cubique

Octaèdres [Clarisó Cortadella 07]

$\bigwedge \sum_i \pm X_i \leq c$

Coût exponentiel en pire cas

Notre contribution : TreeKs

- un foncteur de domaines
- s'applique aux domaines d'inégalités linéaires
- rapport coût/expressivité paramétrable

Notre contribution : TreeKs

- un foncteur de domaines
- s'applique aux domaines d'inégalités linéaires
- rapport coût/expressivité paramétrable

Plan :

- l'opération de complétion
- passer à l'échelle avec les packs
- application de notre foncteur au domaine des zones

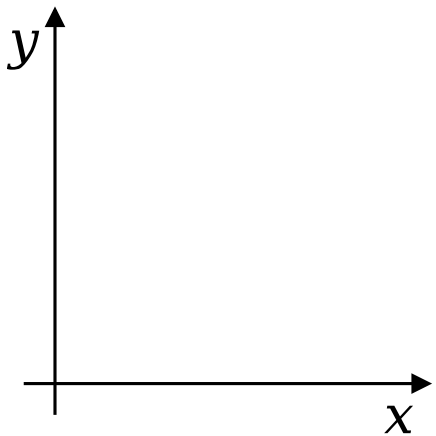
Complétion : une opération fondamentale

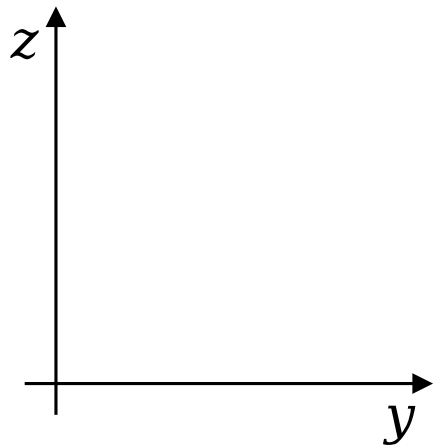
- Point commun des domaines faiblement relationnels
- But : rendre explicite les contraintes implicites
- Se fait par combinaison/propagation de contraintes
- Opération nécessaire aux autres opérations ($\sqcup$, $\sqcap$, $\sqsubseteq$, ...)
- Détermine le coût du domaine

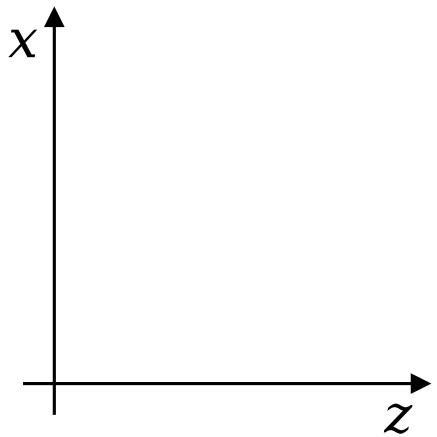
Complétion : exemple

Domaine des zones $(\bigwedge_{ij} X_i - X_j \leq b_{ij})$

$\mathcal{V} = \{x, y, z\}$



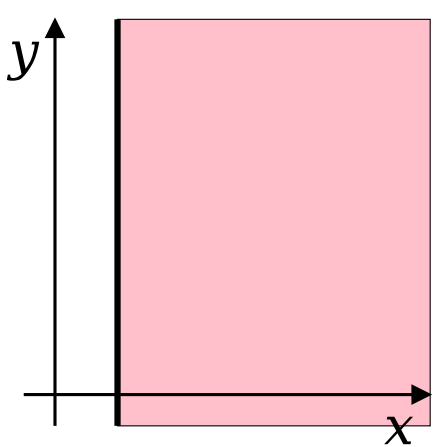


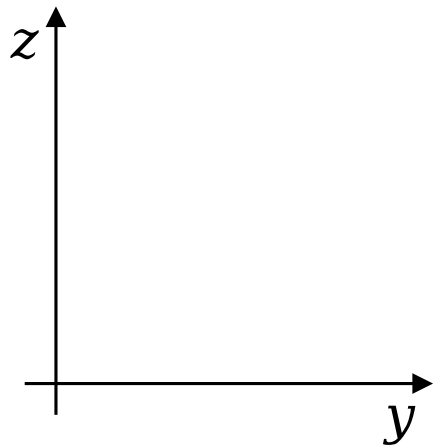


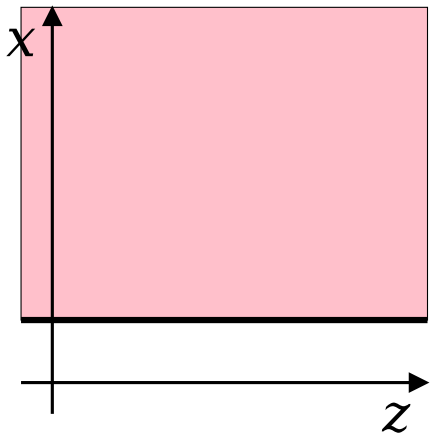
Complétion : exemple

Domaine des zones $(\bigwedge_{ij} X_i - X_j \leq b_{ij})$

$\mathcal{V} = \{x, y, z\}$





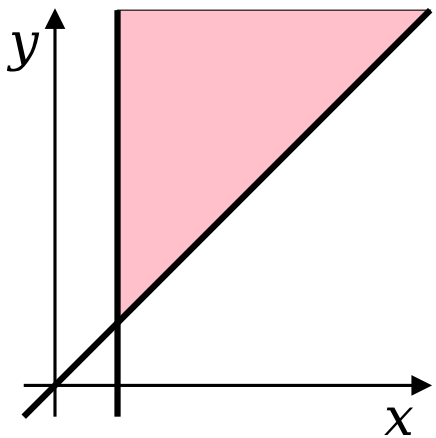


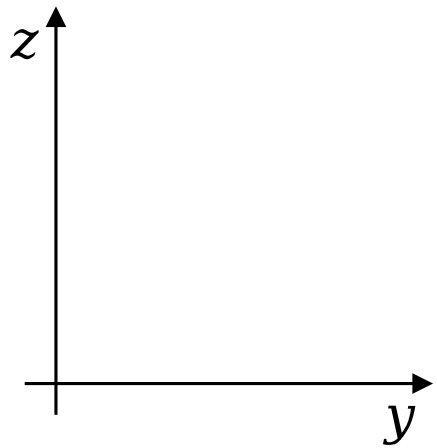
$-x \leq -1$

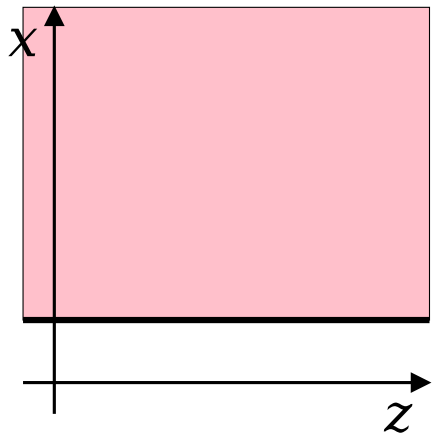
Complétion : exemple

Domaine des zones $(\bigwedge_{ij} X_i - X_j \leq b_{ij})$

$\mathcal{V} = \{x, y, z\}$







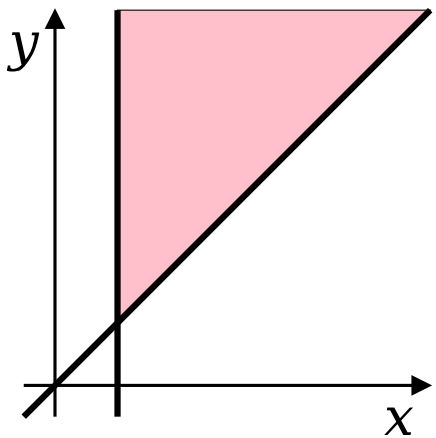
$-x \leq -1$

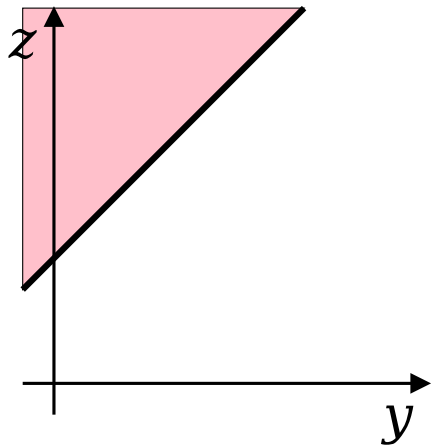
$x - y \leq 0$

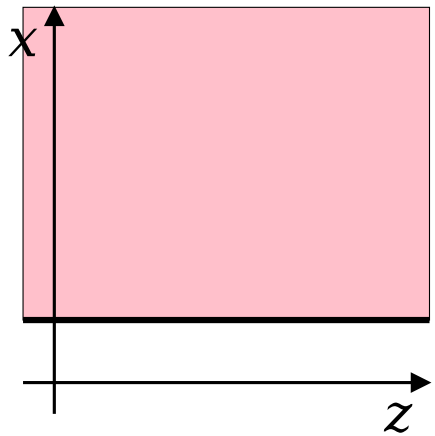
Complétion : exemple

Domaine des zones $(\bigwedge_{ij} X_i - X_j \leq b_{ij})$

$\mathcal{V} = \{x, y, z\}$





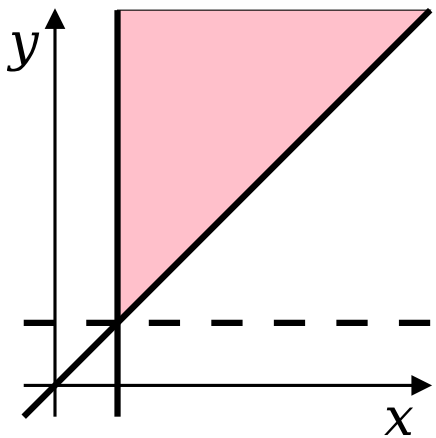


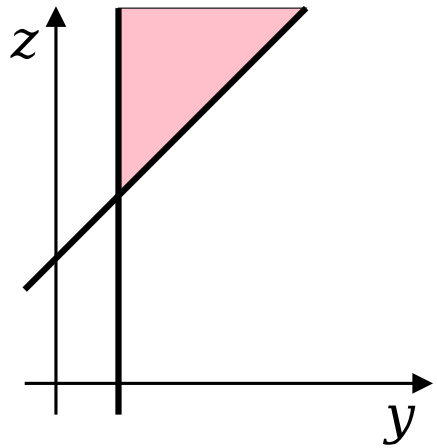
$$-x \leq -1$$
$$x - y \leq 0$$
$$y - z \leq -2$$

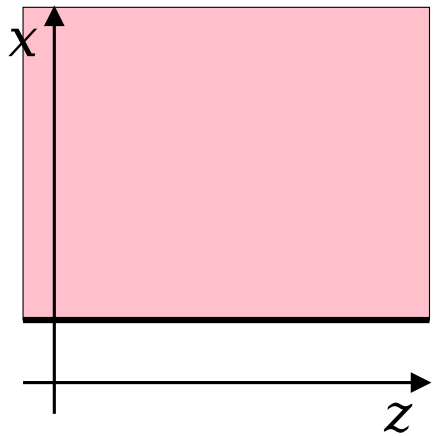
Complétion : exemple

Domaine des zones $(\bigwedge_{ij} X_i - X_j \leq b_{ij})$

$\mathcal{V} = \{x, y, z\}$







$-x \leq -1$

$x - y \leq 0$

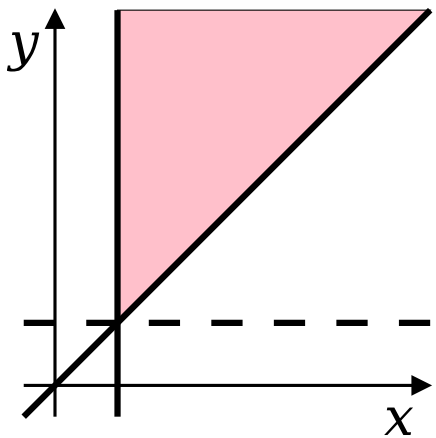
$y - z \leq -2$

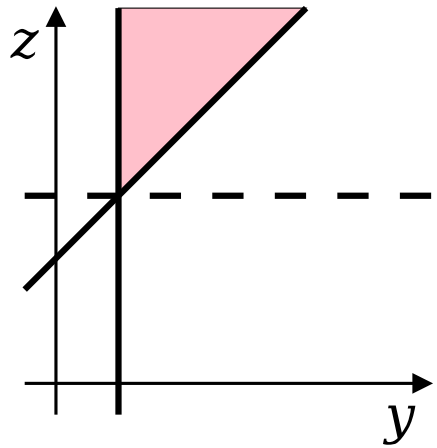
$-y \leq -1$

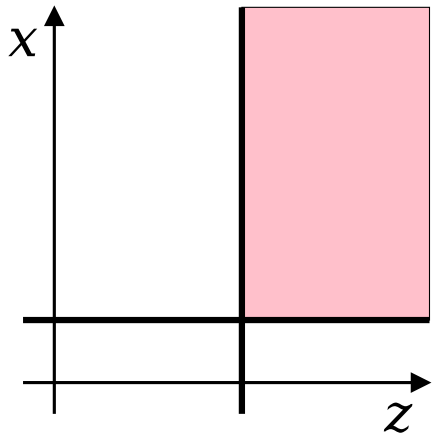
Complétion : exemple

Domaine des zones $(\bigwedge_{ij} X_i - X_j \leq b_{ij})$

$\mathcal{V} = \{x, y, z\}$







$-x \leq -1$

$x - y \leq 0$

$y - z \leq -2$

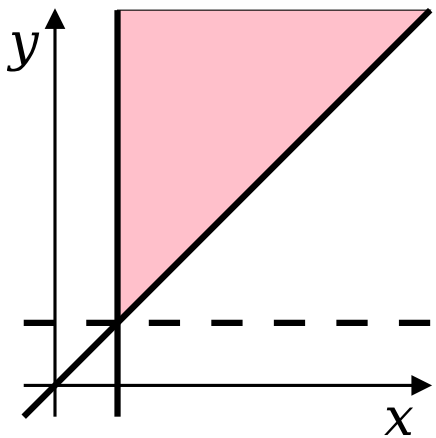
$-y \leq -1$

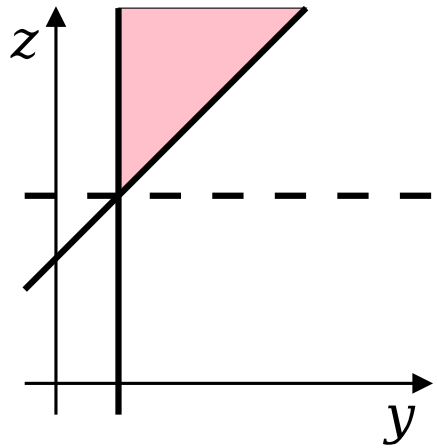
$-z \leq -3$

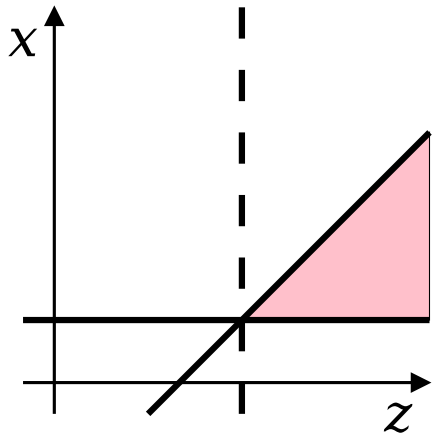
Complétion : exemple

Domaine des zones $(\bigwedge_{ij} X_i - X_j \leq b_{ij})$

$\mathcal{V} = \{x, y, z\}$







$-x \leq -1$

$x - y \leq 0$

$y - z \leq -2$

$-y \leq -1$

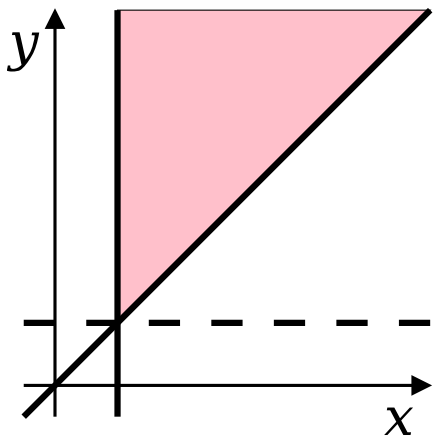
$-z \leq -3$

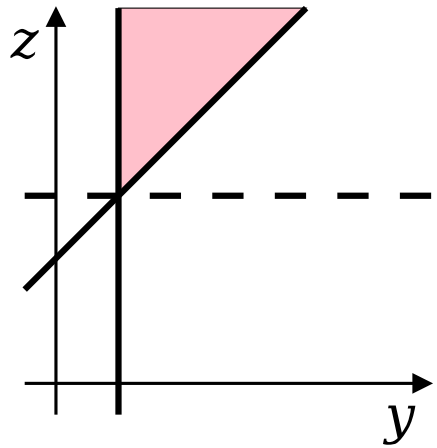
$x - z \leq -2$

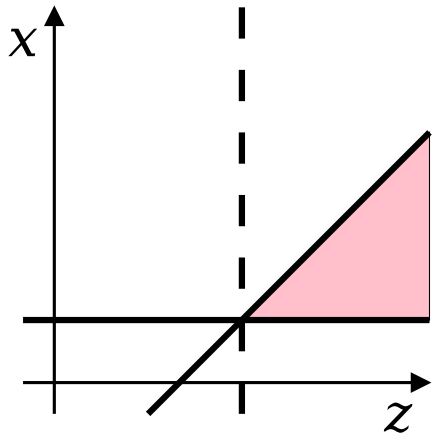
Complétion : exemple

Domaine des zones $(\bigwedge_{ij} X_i - X_j \leq b_{ij})$

$\mathcal{V} = \{x, y, z\}$







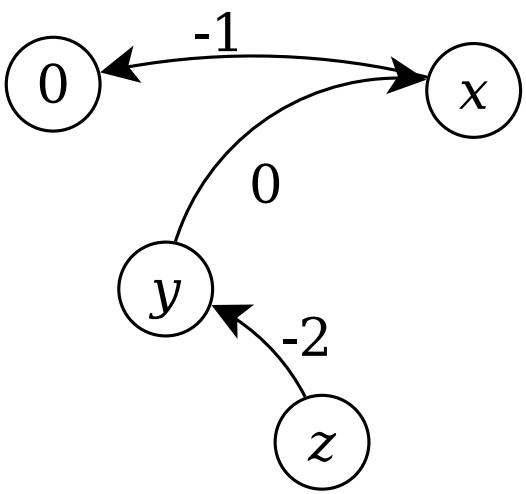
$$-x \leq -1$$
$$x - y \leq 0$$
$$y - z \leq -2$$

$$-y \leq -1$$
$$-z \leq -3$$
$$x - z \leq -2$$

Terminé !

Domaine des zones : représentation

On représente un ensemble de contraintes de différences de deux variables ($X_i - X_j \leq \mathbf{m}_{ji}$) par un graphe de potentiel ou par une DBM (*Difference Bound Matrice*).

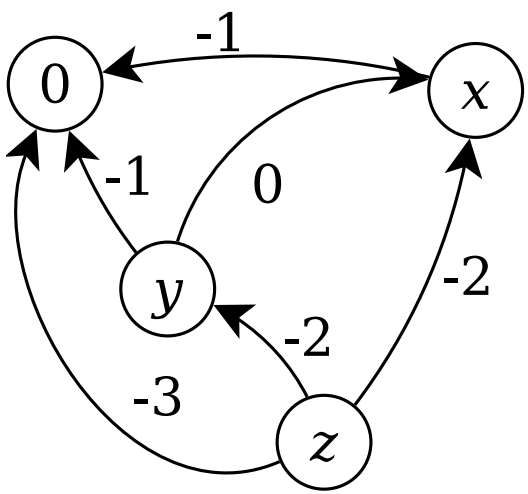


	0	x	y	z
0	0	$+\infty$	$+\infty$	$+\infty$
x	-1	0	$+\infty$	$+\infty$
y	$+\infty$	0	0	$+\infty$
z	$+\infty$	$+\infty$	-2	0

$$
\begin{aligned}
0 - x &\leq -1 \\
x - y &\leq 0 \\
y - z &\leq -2
\end{aligned}
$$

Domaine des zones : représentation

On représente un ensemble de contraintes de différences de deux variables ($X_i - X_j \leq \mathbf{m}_{ji}$) par un graphe de potentiel ou par une DBM (*Difference Bound Matrice*).



	0	x	y	z
0	0	$+\infty$	$+\infty$	$+\infty$
x	-1	0	$+\infty$	$+\infty$
y	-1	0	0	$+\infty$
z	-3	-2	-2	0

$$\begin{aligned} 0 - x &\leq -1 \\ x - y &\leq 0 \\ y - z &\leq -2 \end{aligned}$$

$$\begin{aligned} 0 - y &\leq -1 \\ 0 - z &\leq -3 \\ x - z &\leq -2 \end{aligned}$$

Domaine des zones : complétion

Dans le domaine des zones, l'opération de complétion est une clotûre de plus courts chemins.

Algorithme de Floyd-Warshall $O(n^3)$

---

pour $k \leftarrow 1$ **à** N **faire**
  pour $i \leftarrow 1$ **à** N **faire**
    pour $j \leftarrow 1$ **à** N **faire**
      $\mathbf{m}_{ij} \leftarrow \min(\mathbf{m}_{ij}, \mathbf{m}_{ik} + \mathbf{m}_{kj})$

À la fin : $\begin{cases} \forall i, j, k, \mathbf{m}_{ij} \leq \mathbf{m}_{ik} + \mathbf{m}_{kj} & \text{si satisfiable} \\ \exists i, \mathbf{m}_{ii} < 0 & \text{si insatisfiable} \end{cases}$

Domaine des zones : opérateurs

Sur des valeurs complètes, les opérations se font point-à-point.

Jointure (meilleure approximation de l'union) :

$$(\mathbf{m} \sqcup \mathbf{n})_{ij} = \max(\mathbf{m}_{ij}, \mathbf{n}_{ij})$$

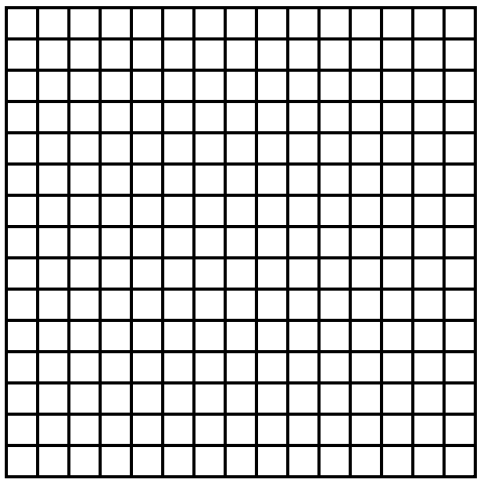
Opérateur d'oubli (projection) :

$$(\exists_{X_k} \mathbf{m})_{ij} = \begin{cases} \mathbf{m}_{ij} & \text{si } i \neq k \text{ et } j \neq k \\ 0 & \text{si } i = j = k \\ +\infty & \text{sinon} \end{cases}$$

Passer à l'échelle : les packs d'Astrée

Principe :

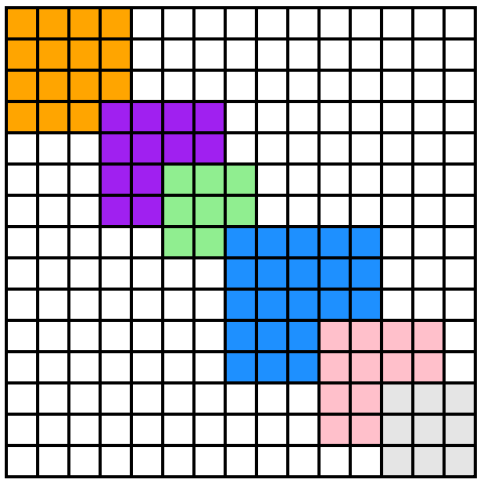
- répartir les variables dans des packs
- utiliser une DBM par pack



Passer à l'échelle : les packs d'Astrée

Principe :

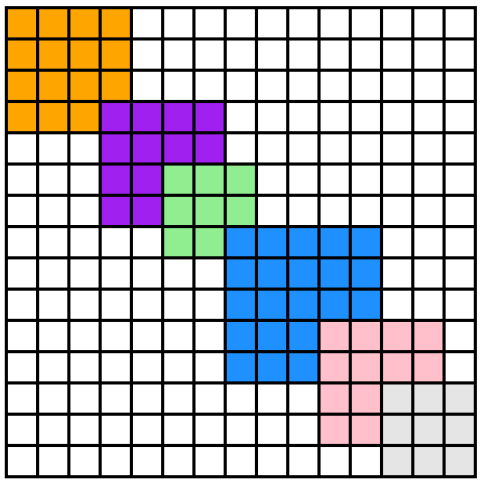
- répartir les variables dans des packs
- utiliser une DBM par pack

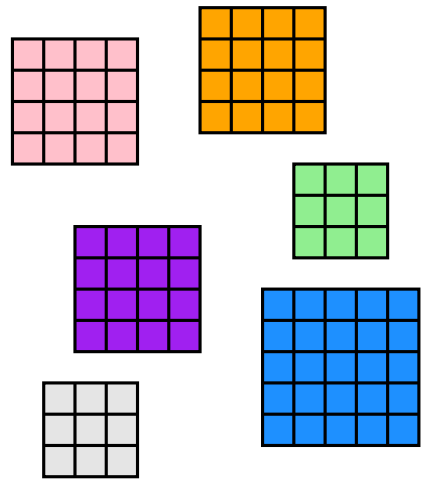


Passer à l'échelle : les packs d'Astrée

Principe :

- répartir les variables dans des packs
- utiliser une DBM par pack

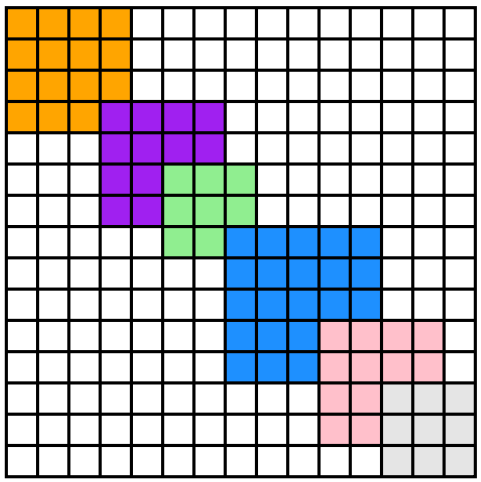


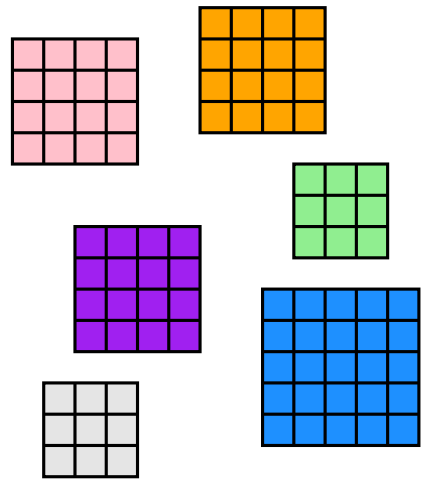


Passer à l'échelle : les packs d'Astrée

Principe :

- répartir les variables dans des packs
- utiliser une DBM par pack



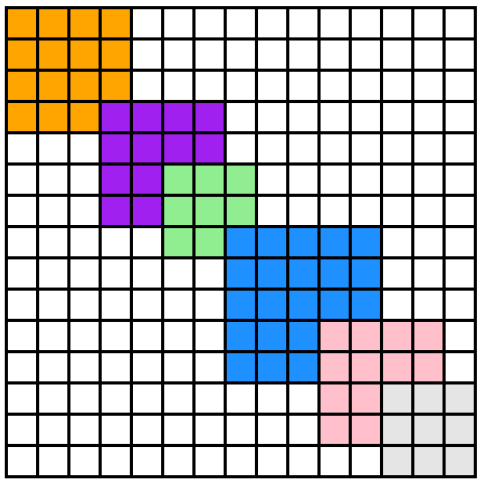


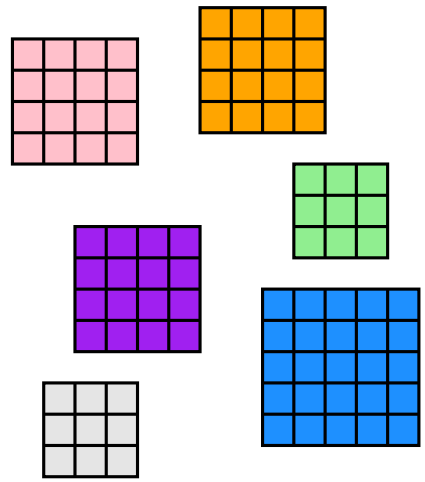
Coût : linéaire pour des packs de taille bornée

Passer à l'échelle : les packs d'Astrée

Principe :

- répartir les variables dans des packs
- utiliser une DBM par pack





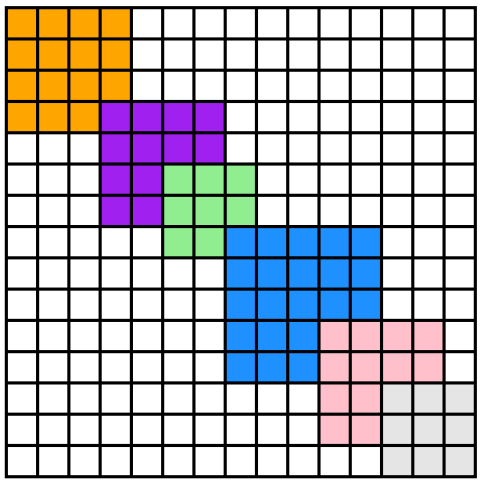
Coût : linéaire pour des packs de taille bornée

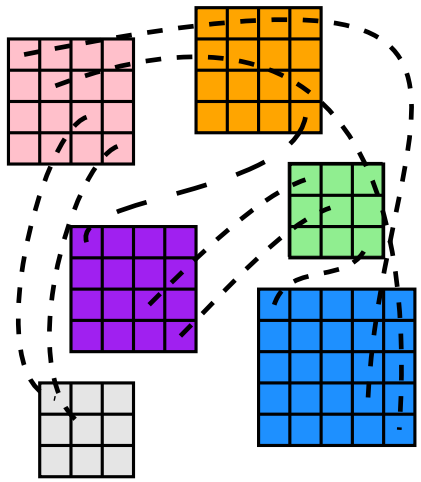
Perte d'information : pas de communication entre les packs !

Passer à l'échelle : les packs d'Astrée

Principe :

- répartir les variables dans des packs
- utiliser une DBM par pack





Coût : linéaire pour des packs de taille bornée

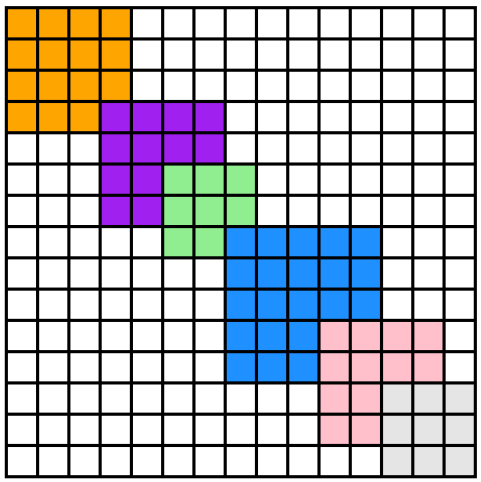
Perte d'information : pas de communication entre les packs !

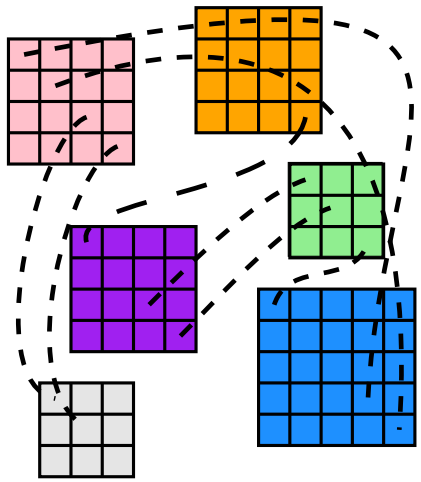
Solution : partage des contraintes d'intervalles

Passer à l'échelle : les packs d'Astrée

Principe :

- répartir les variables dans des packs
- utiliser une DBM par pack





Coût : linéaire pour des packs de taille bornée

Perte d'information : pas de communication entre les packs !

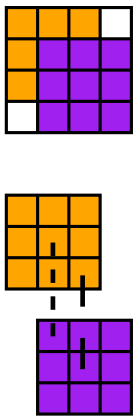
Solution : partage des contraintes d'intervalles

Précision insuffisante !

Passer à l'échelle : les packs d'Astrée

Principe :

- répartir les variables dans des packs
- utiliser une DBM par pack



$$P_1 = \{t, x, y\} \qquad P_2 = \{t, x, z\}$$

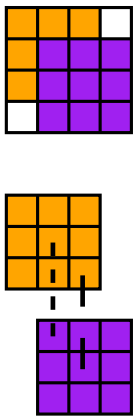
$$t \leq y \qquad x \leq z$$
$$y \leq x \qquad z \leq t$$

Coût : linéaire pour des packs de taille bornée
Perte d'information : pas de communication entre les packs !
Solution : partage des contraintes d'intervalles
Précision insuffisante !

Passer à l'échelle : les packs d'Astrée

Principe :

- répartir les variables dans des packs
- utiliser une DBM par pack



$$P_1 = \{t, x, y\} \qquad P_2 = \{t, x, z\}$$

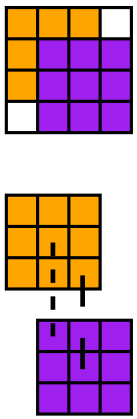
$$t \leq y \qquad x \leq z$$
$$y \leq x \qquad z \leq t$$
$$t \leq x \qquad x \leq t$$

Coût : linéaire pour des packs de taille bornée
Perte d'information : pas de communication entre les packs !
Solution : partage des contraintes d'intervalles
Précision insuffisante !

Passer à l'échelle : les packs d'Astrée

Principe :

- répartir les variables dans des packs
- utiliser une DBM par pack



$$P_1 = \{t, x, y\} \qquad P_2 = \{t, x, z\}$$

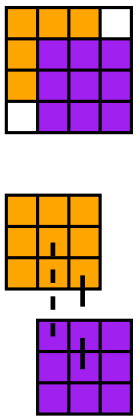
$$t \leq y \qquad x \leq z$$
$$y \leq x \qquad z \leq t$$
$$t \leq x \qquad x \leq t$$

Coût : linéaire pour des packs de taille bornée
Perte d'information : pas de communication entre les packs !
Solution : partage des contraintes d'intervalles
Précision insuffisante !

Passer à l'échelle : les packs d'Astrée

Principe :

- répartir les variables dans des packs
- utiliser une DBM par pack



$$P_1 = \{t, x, y\} \qquad P_2 = \{t, x, z\}$$

$$t \leq y \qquad x \leq z$$
$$y \leq x \qquad z \leq t$$
$$t \leq x \qquad x \leq t$$

$$x = t$$

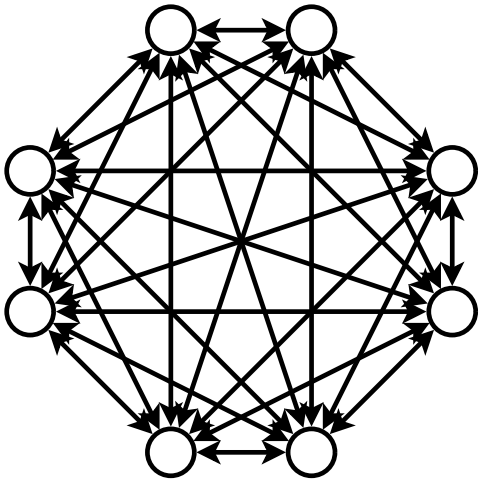
Coût : linéaire pour des packs de taille bornée
Perte d'information : pas de communication entre les packs !
Solution : partage des contraintes d'intervalles
Précision insuffisante !

Idée : un sous-graphe

But : partager les contraintes relationnelles

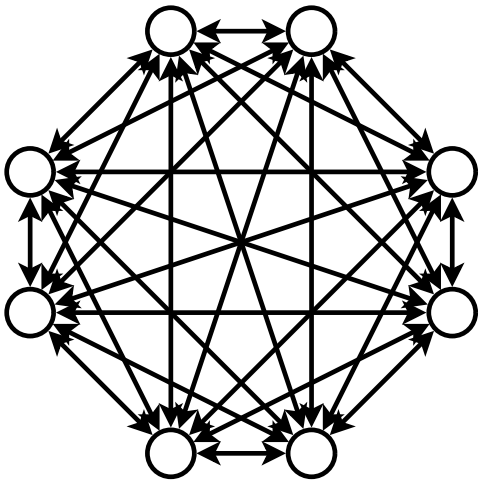
Idée : un sous-graphe

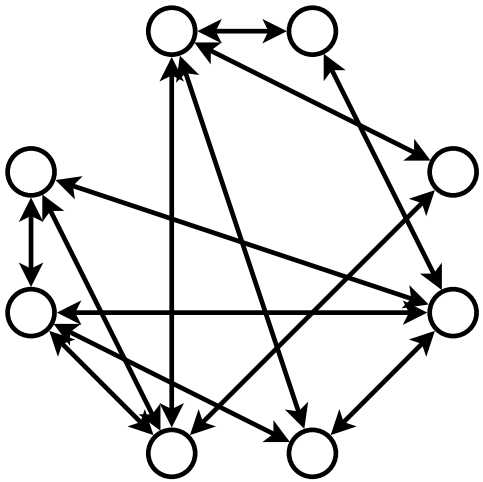
But : partager les contraintes relationnelles

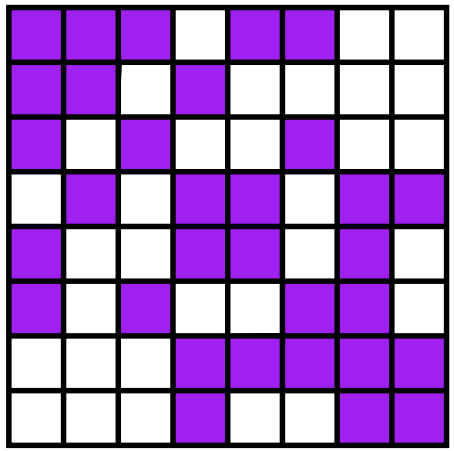


Idée : un sous-graphe

But : partager les contraintes relationnelles

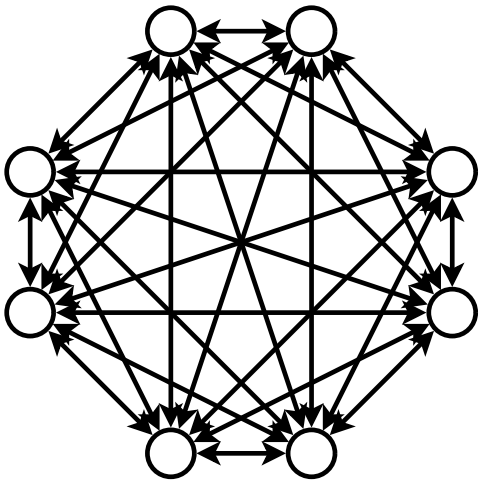


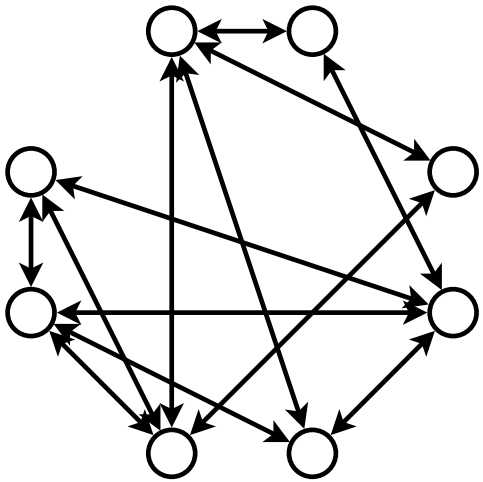


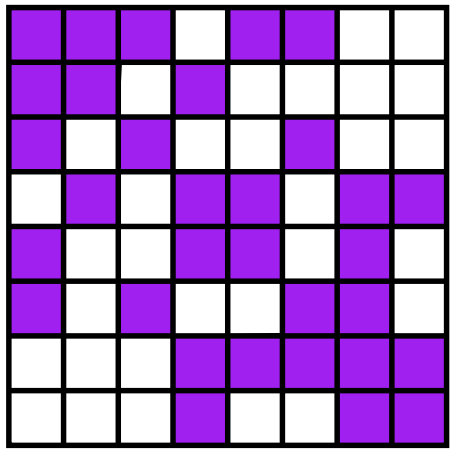


Idée : un sous-graphe

But : partager les contraintes relationnelles



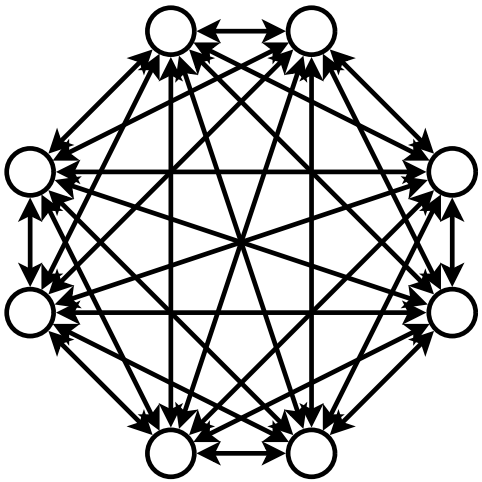


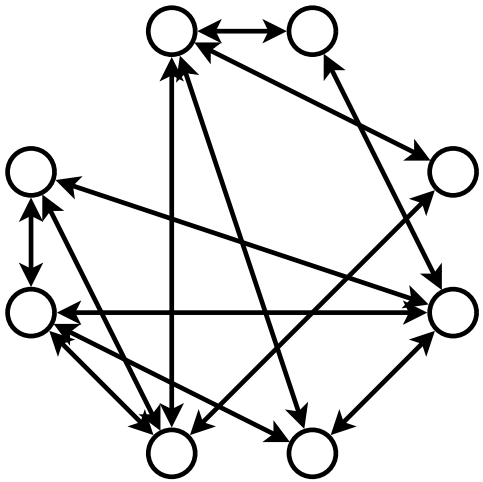


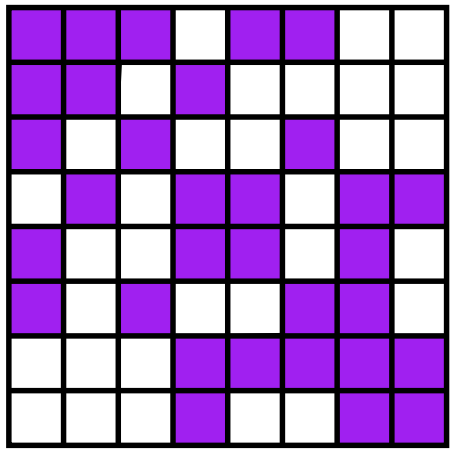
Problèmes :

Idée : un sous-graphe

But : partager les contraintes relationnelles







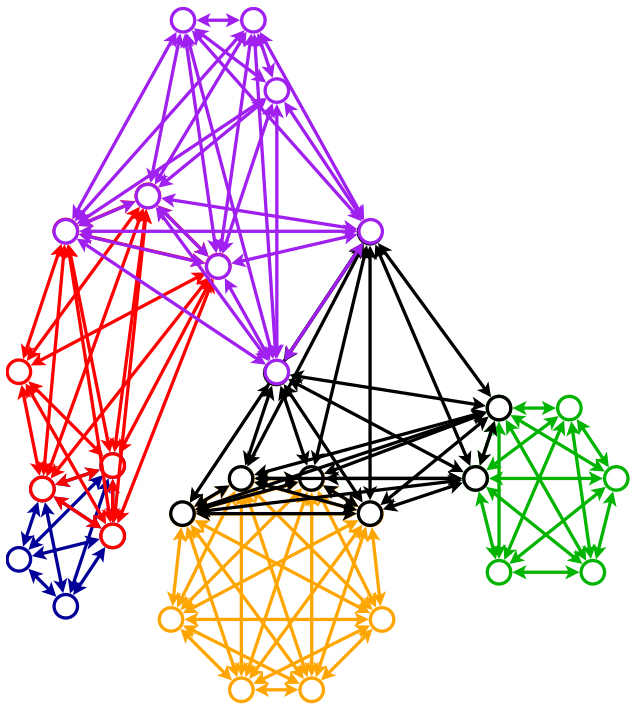
Problèmes :

- conserver une bonne expressivité
- conserver une structure de packs
- conserver des algorithmes précis et efficaces

TreeKs : un certain sous-graphe

Forme :

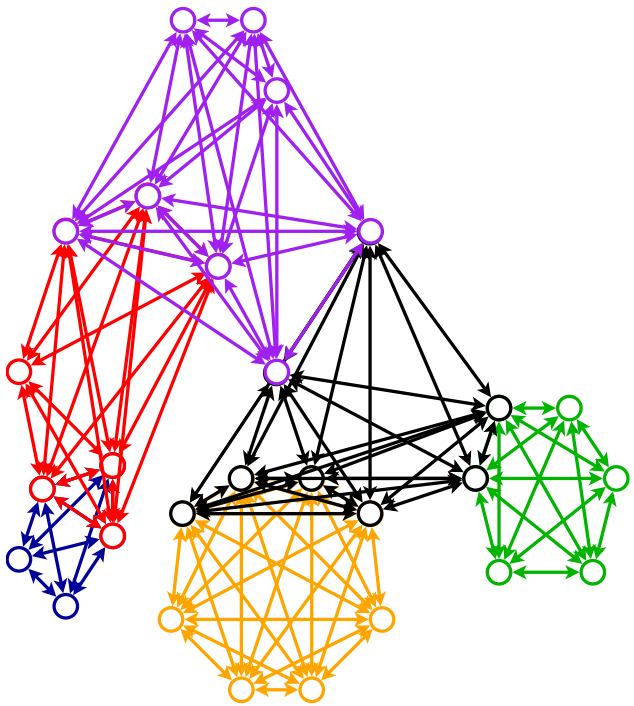
- un arbre de graphes complets (packs)
- qui partagent des frontières communes

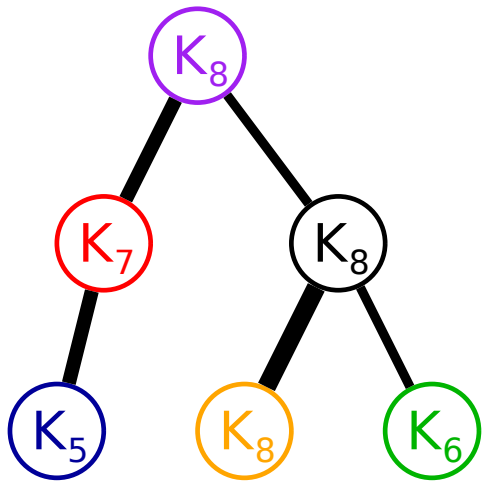


TreeKs : un certain sous-graphe

Forme :

- un arbre de graphes complets (packs)
- qui partagent des frontières communes



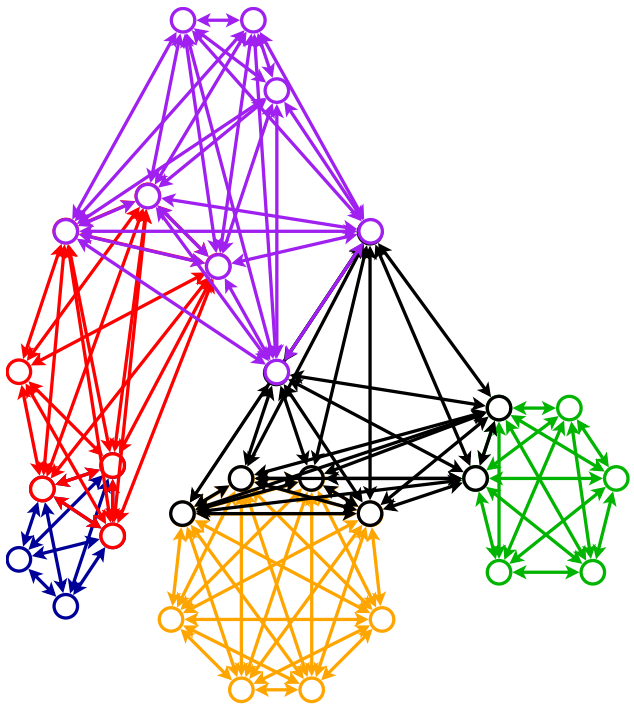


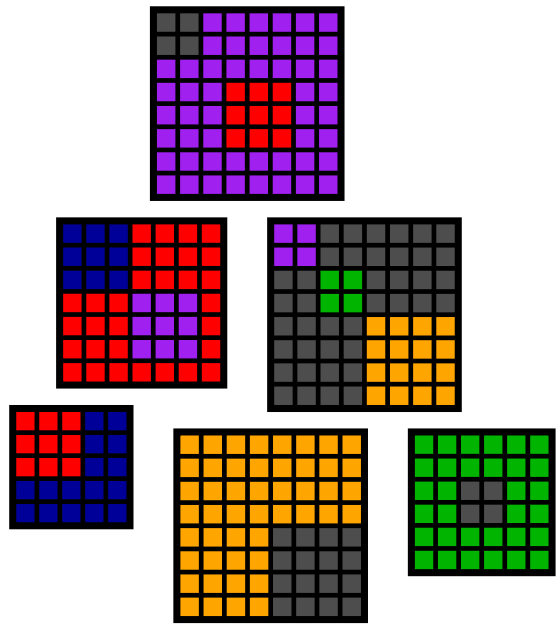
arbre des packs

TreeKs : un certain sous-graphe

Forme :

- un arbre de graphes complets (packs)
- qui partagent des frontières communes



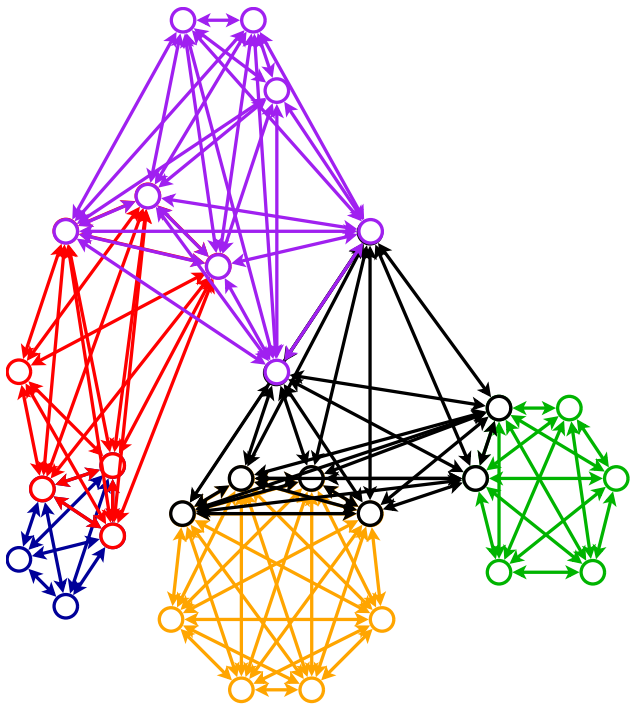


Valeur abstraite : n-uplet de DBMs

TreeKs : un certain sous-graphe

Forme :

- un arbre de graphes complets (packs)
- qui partagent des frontières communes



Paramètres :

N	nombre de variables
m	nombre de packs
p	taille d'un pack
f	taille d'une frontière
d	diamètre du graphe

TreeKs : opérateurs abstraits

Sur des valeurs complètes, les opérations se font pack à pack :

- test d'inclusion
- intersection
- union

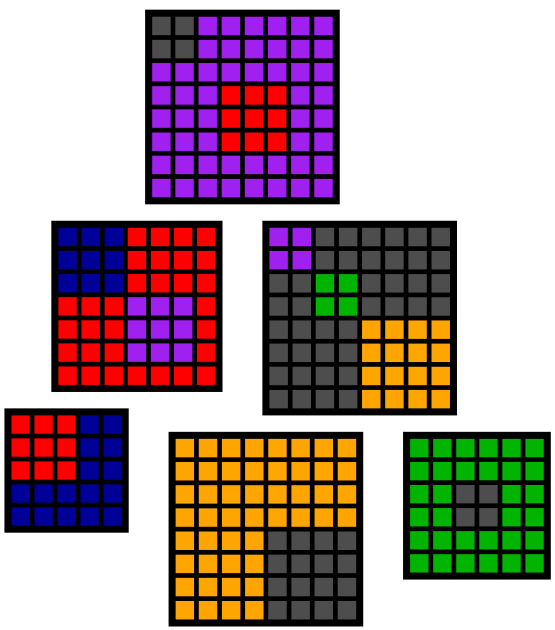
mais pas l'extraction de contraintes, ni l'ajout de contraintes. . .

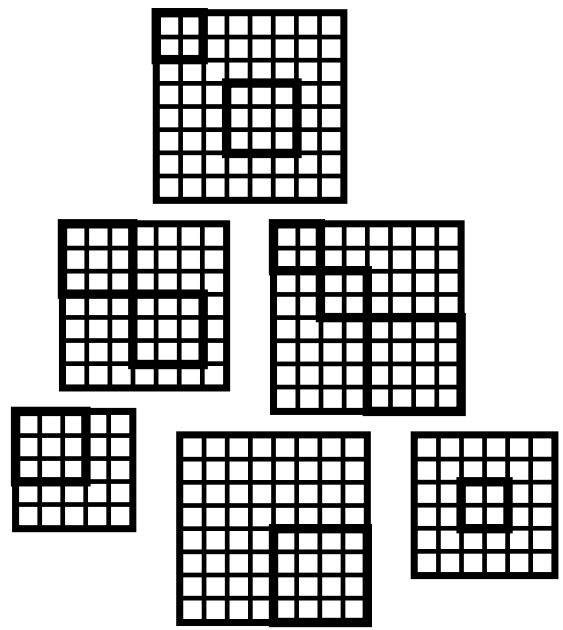
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



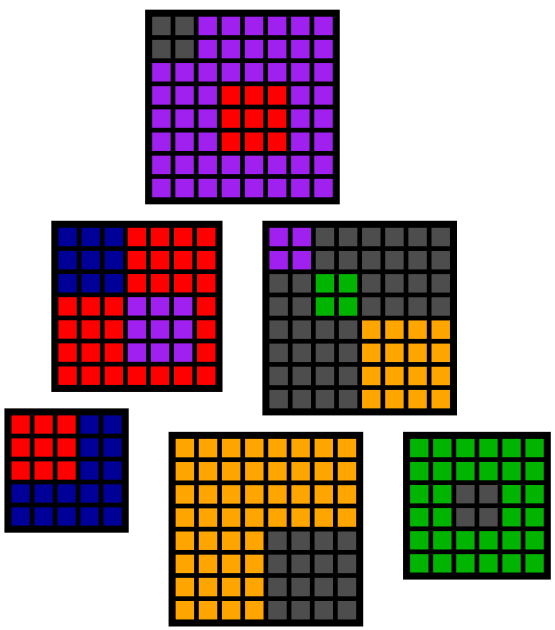


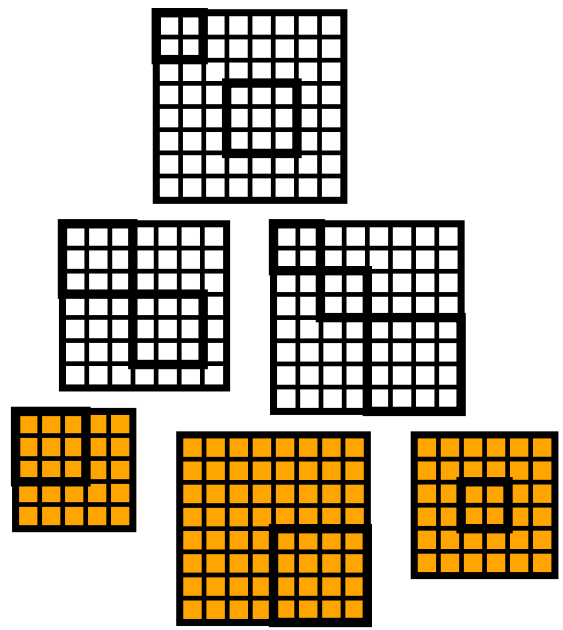
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



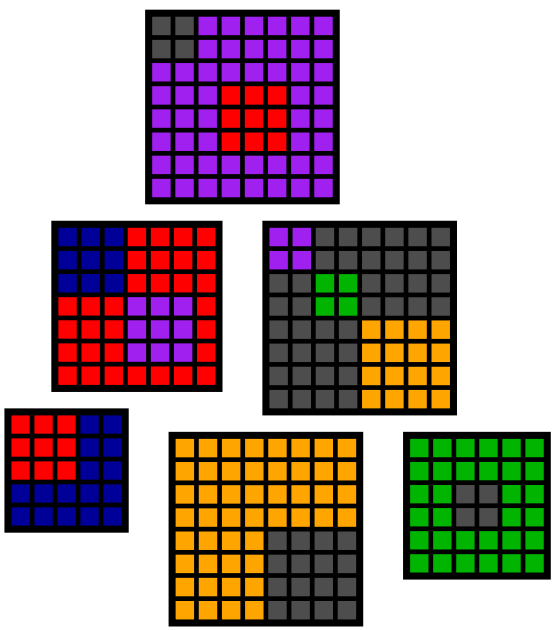


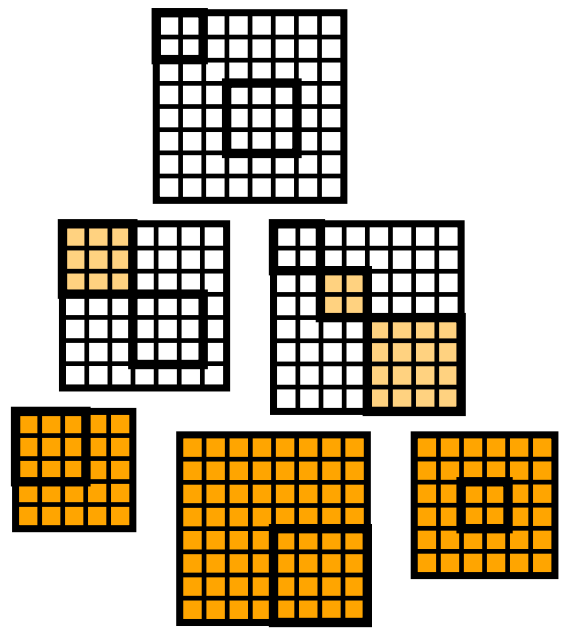
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



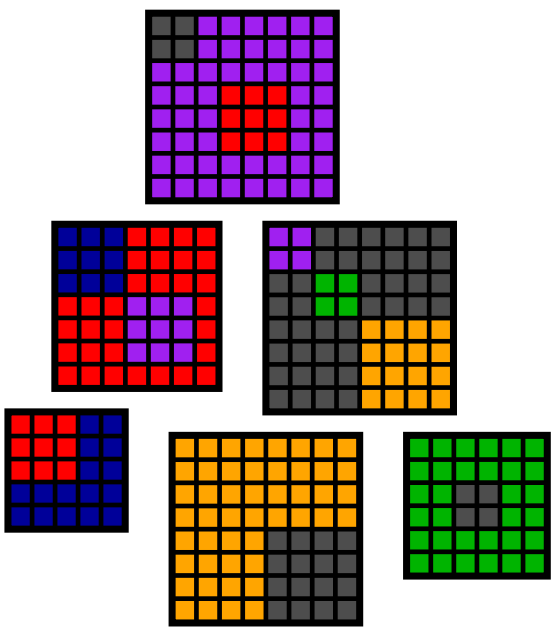


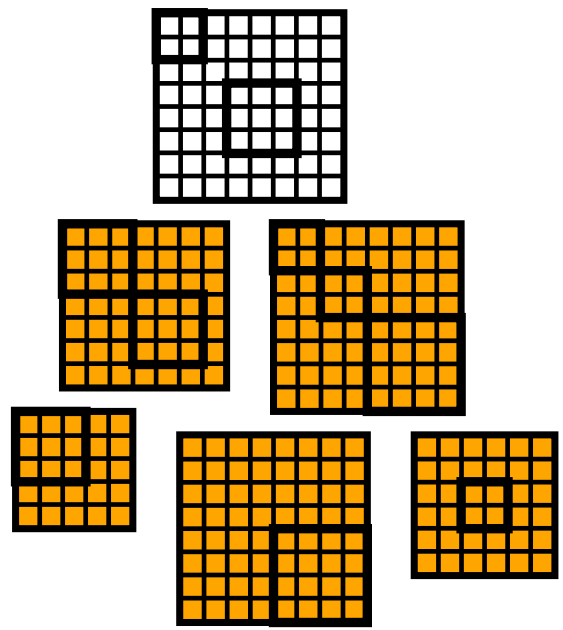
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



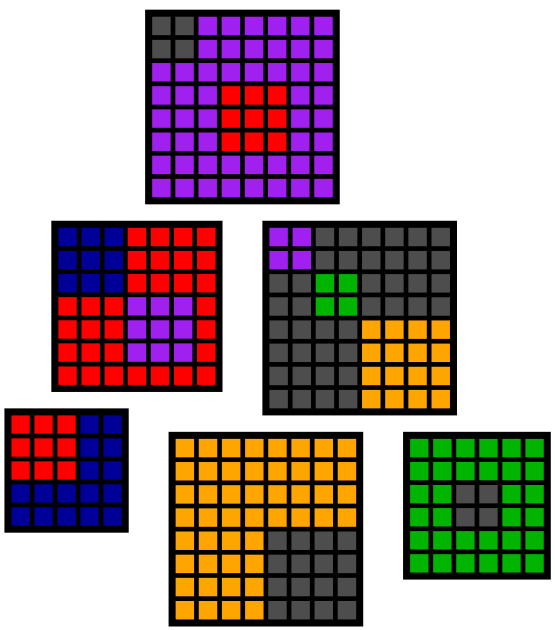


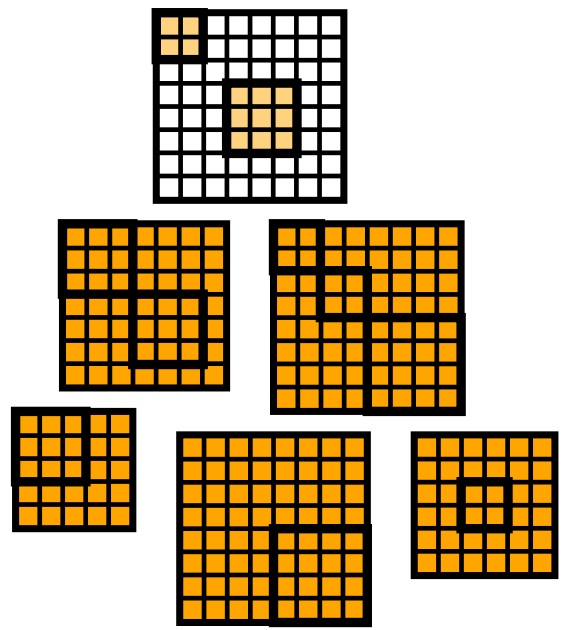
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



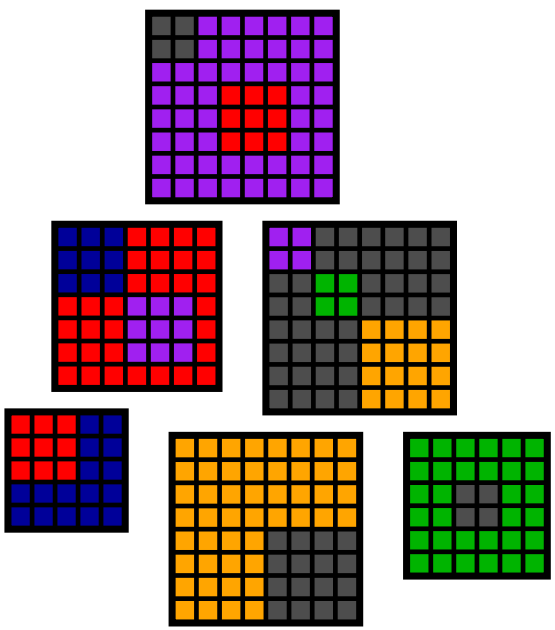


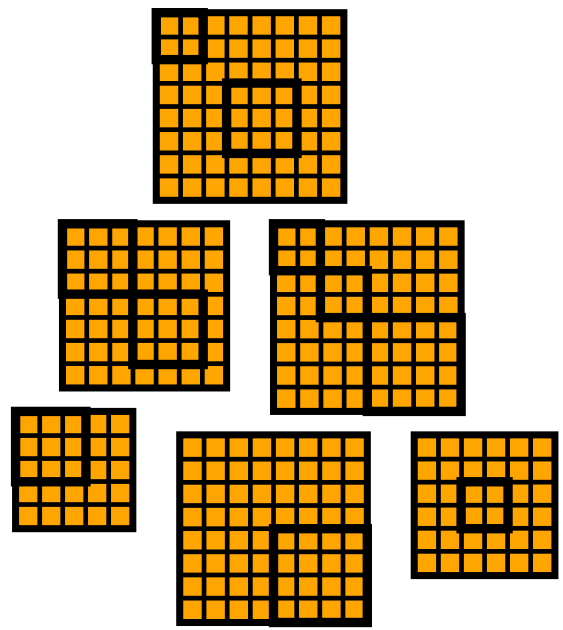
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



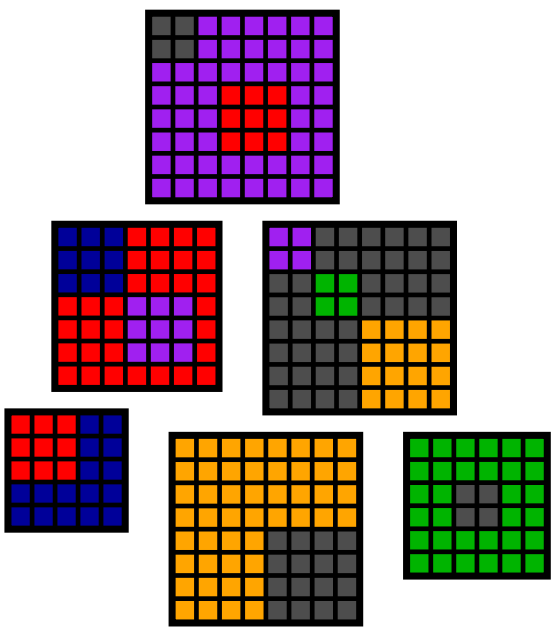


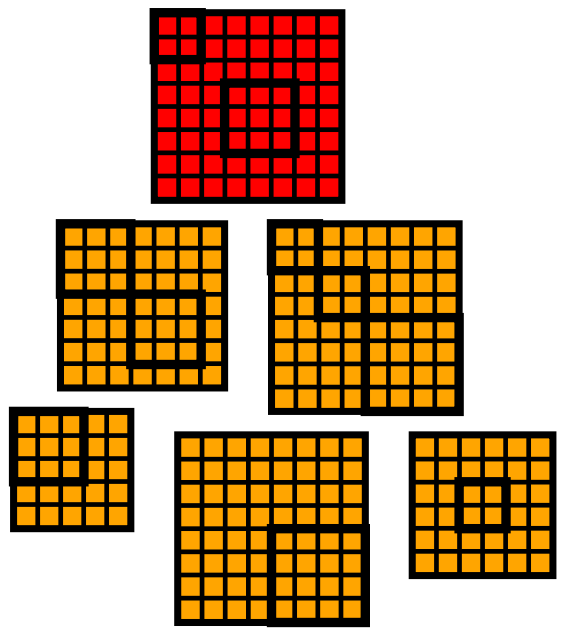
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



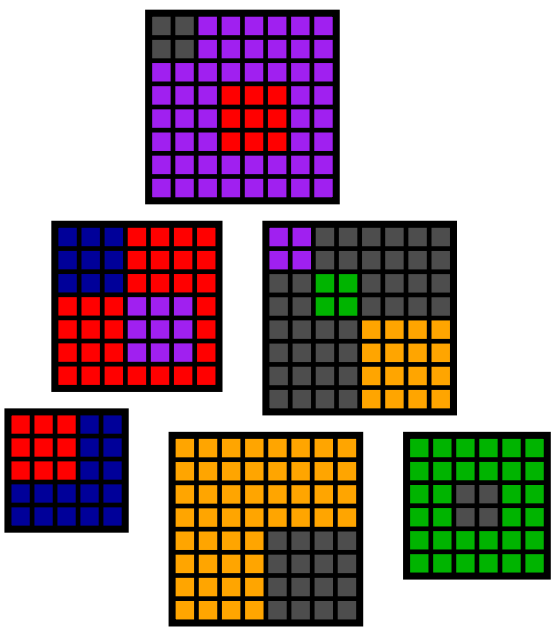


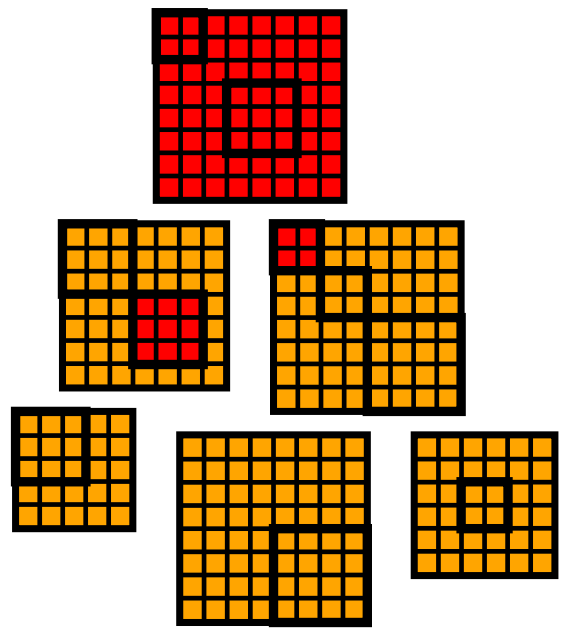
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



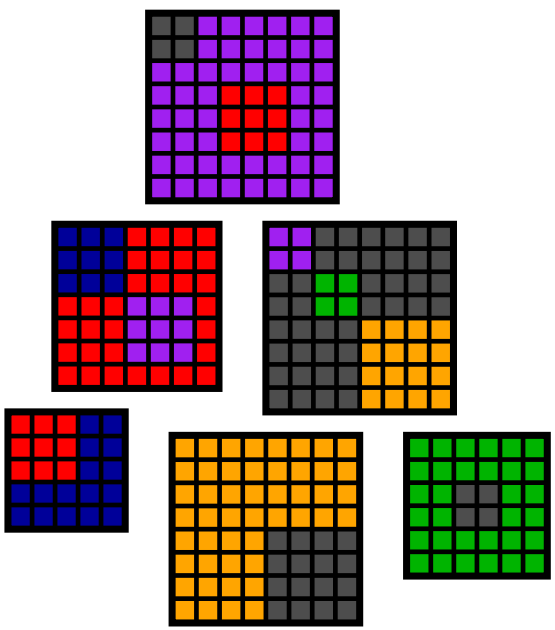


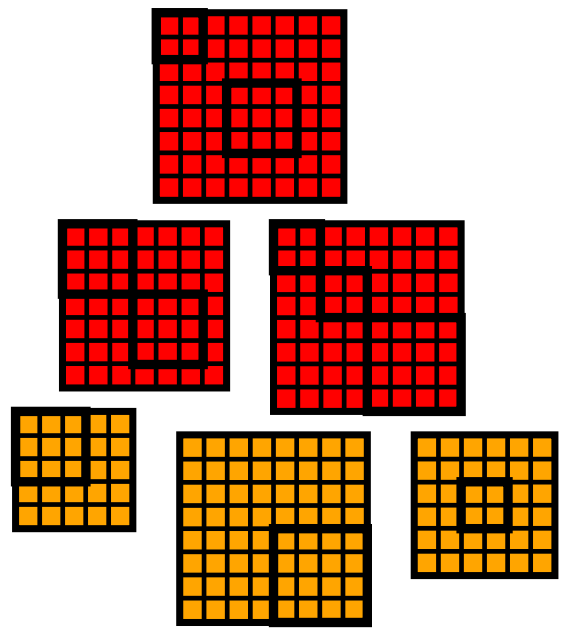
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



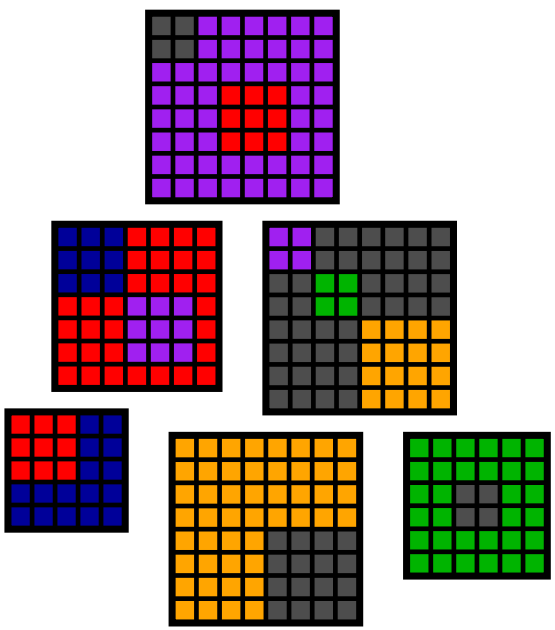


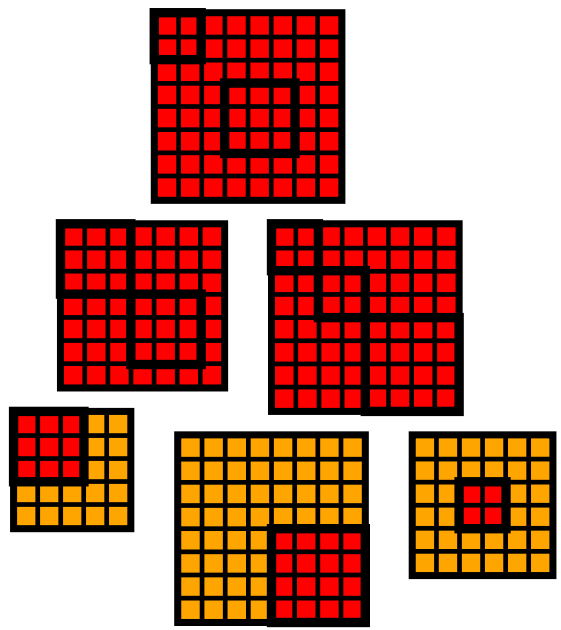
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants



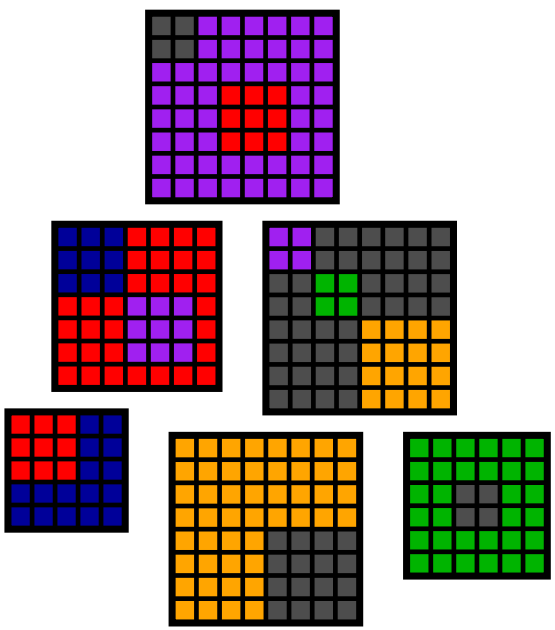


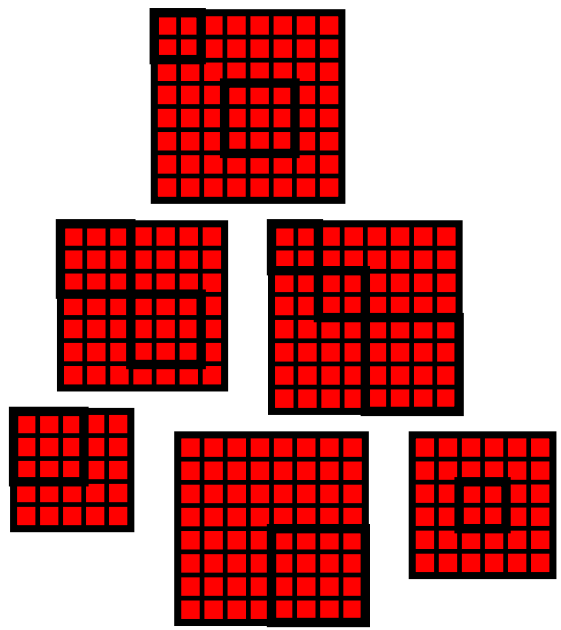
Algorithme de complétion

Algorithme de complétion dans TreeKs $O(mp^3)$

pour chaque *pack des feuilles vers la racine*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à son père

pour chaque *pack de la racine vers les feuilles*
- Compléter ce pack dans le domaine des zones
- Transmettre les nouvelles contraintes à ses enfants

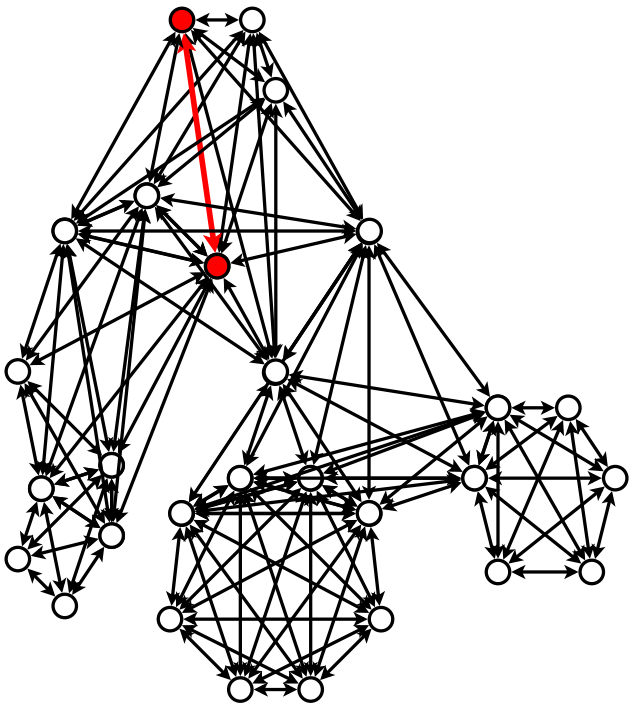


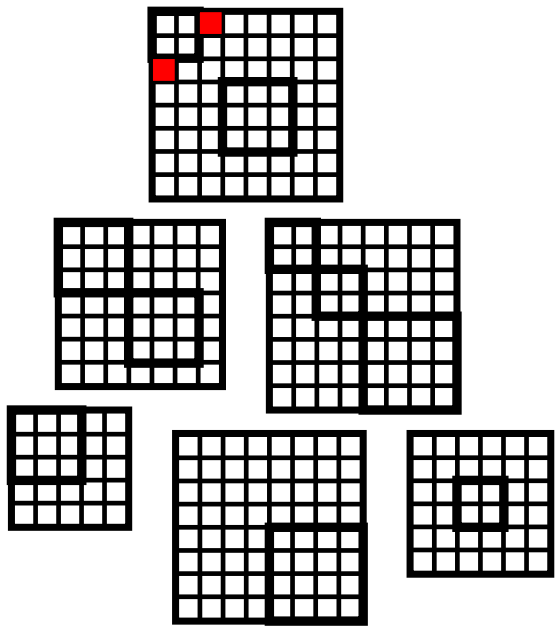


Extraction de contraintes

But : on cherche à borner $X_u - X_v$

Cas simple : X_u et X_v sont dans le même pack

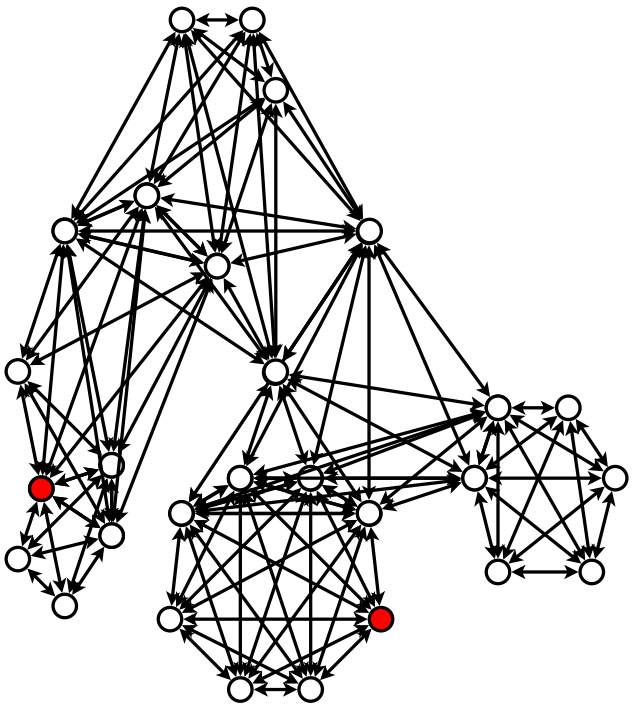


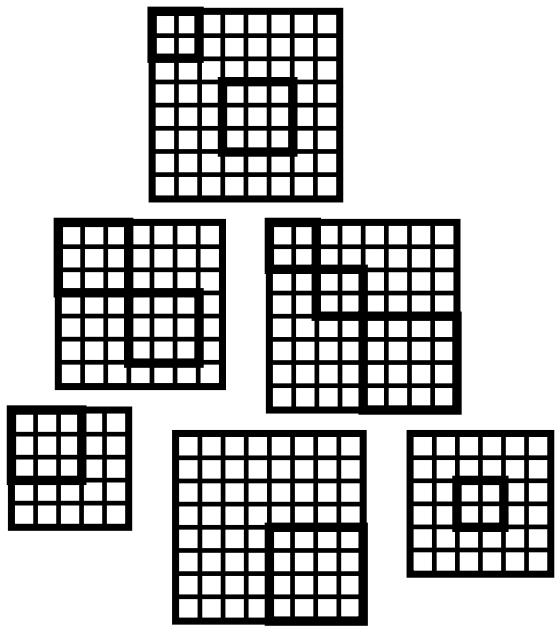


Extraction de contraintes

But : on cherche à borner $X_u - X_v$

Cas complexe : X_u et X_v sont dans des packs différents

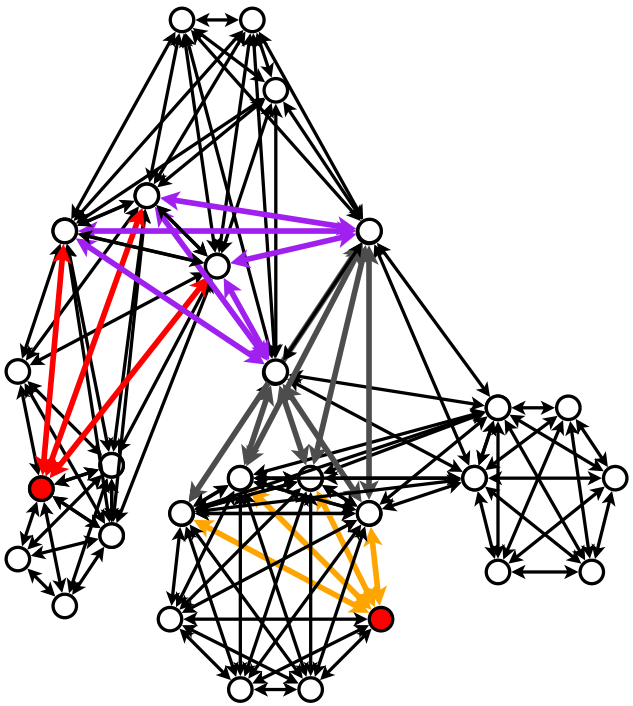


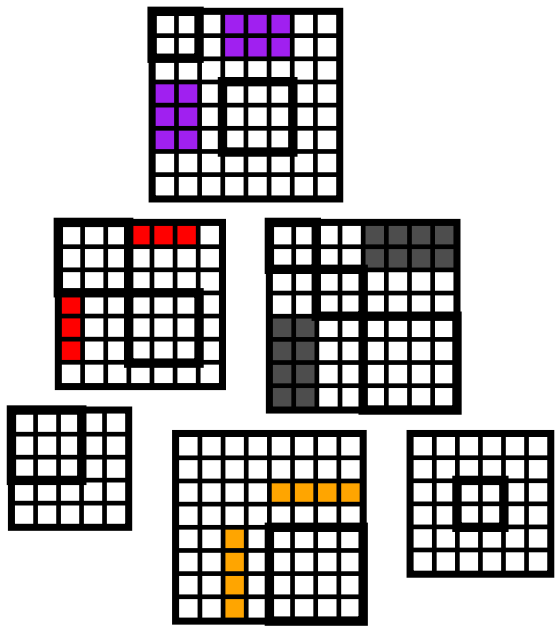


Extraction de contraintes

But : on cherche à borner $X_u - X_v$

Cas complexe : X_u et X_v sont dans des packs différents



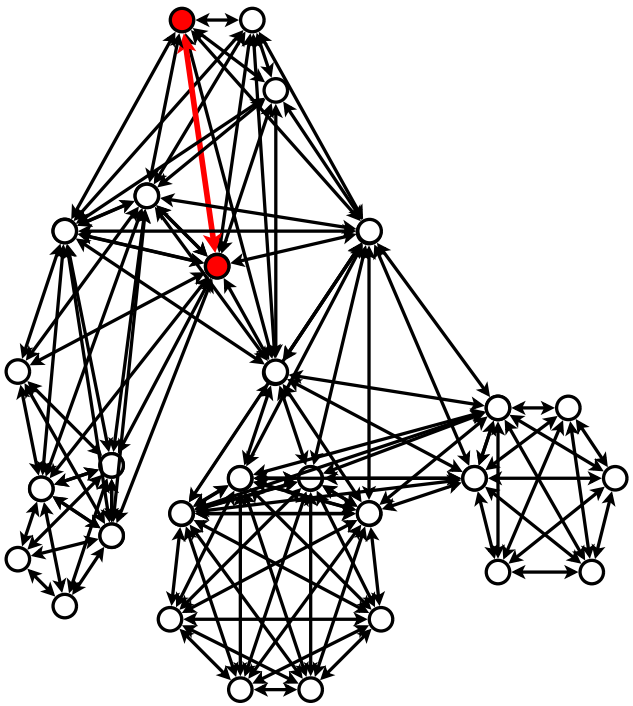


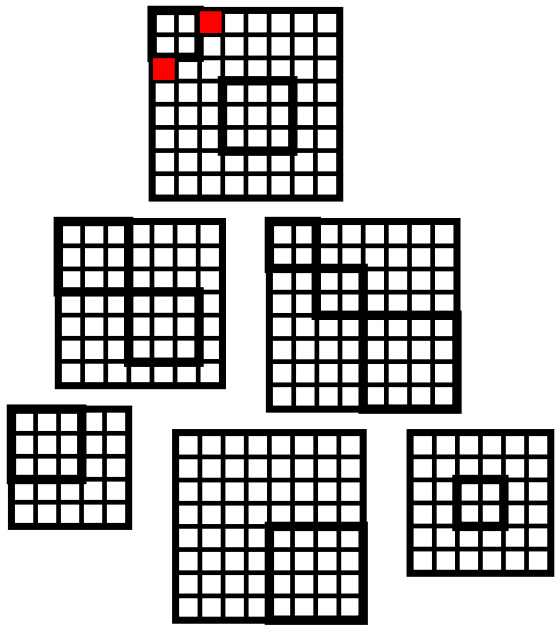
Solution : algorithme dynamique en $O(df^2)$

Ajout de contraintes

But : on souhaite ajouter la contrainte $X_u - X_v \leq c$

Cas simple : X_u et X_v sont dans le même pack

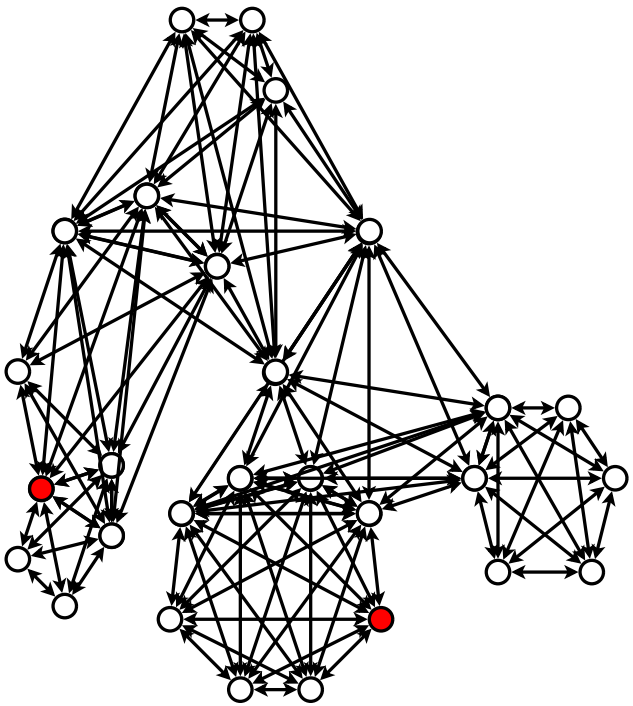


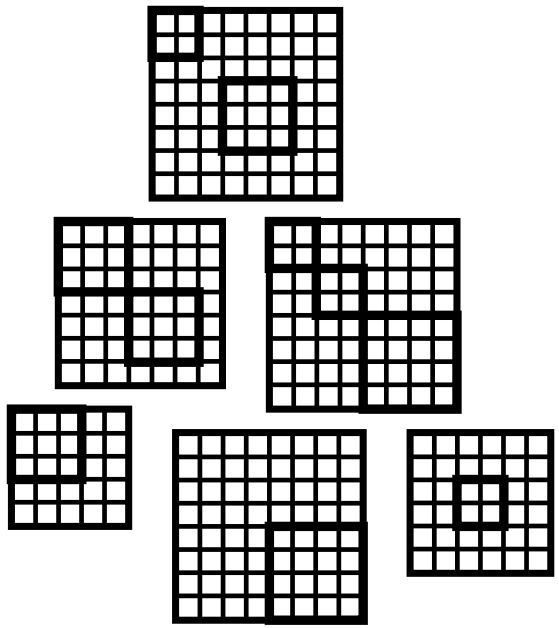


Ajout de contraintes

But : on souhaite ajouter la contrainte $X_u - X_v \leq c$

Cas complexe : X_u et X_v sont dans des packs différents

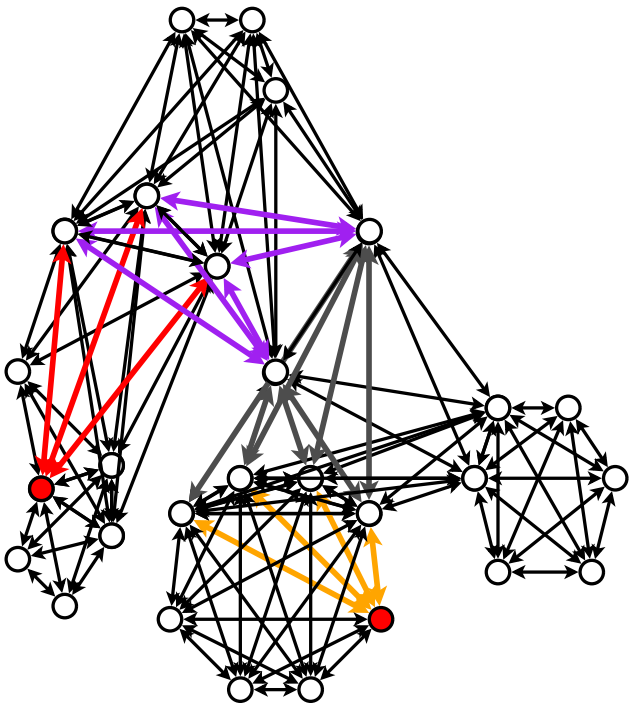


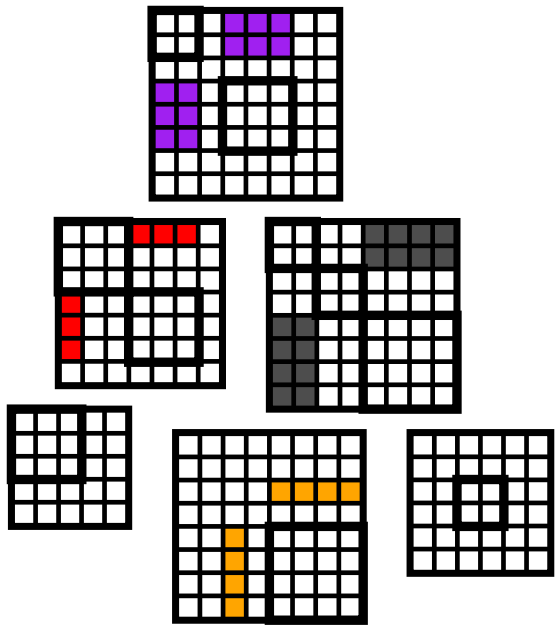


Ajout de contraintes

But : on souhaite ajouter la contrainte $X_u - X_v \leq c$

Cas complexe : X_u et X_v sont dans des packs différents





Pour chaque arc entre X_u et X_v, ajouter la meilleure contrainte qui se déduit de $X_u - X_v \leq c$ et des contraintes existantes : $O(df^2)$

Conclusion

- s'applique à une grande famille de domaines numériques (zones, octogones, logaèdres, TVPI, octaèdres, polyèdres, ...)
- s'applique à des domaines qui pourront être inventés à l'avenir
- pour une taille de packs fixée, la complétion est linéaire
- les algorithmes sont simples, précis et efficaces

Travaux futurs :

- développement de stratégies de génération de packs
- application à d'autres domaines convexes et à des domaines non convexes

Merci pour votre attention

Les questions sont les bienvenues